

Présentation d'une métaheuristique : Problème du voyageur de commerce avec fenêtre de temps

Pierrick Bournez et Julien Koenig

Janvier 2023

1 Définition du problème

Le problème du voyageur de commerce (que l'on nommera par la suite *TSP* en référence à l'anglais *Travelling Salesman Problem*) consiste à trouver dans un graphe complet pondéré le chemin le plus court passant par tous les sommets du graphe une et une seule fois. Le problème porte ce nom car la solution peut être utilisée par un voyageur de commerce devant visiter le plus rapidement possible un certain nombre de villes.

On rajoute ici une contrainte qui est utile dans la pratique (par exemple si le voyageur prend le train) sur l'horaire durant lequel le voyageur peut être présent dans la ville. Il existe donc pour chaque sommet une fenêtre de temps durant laquelle on doit y être. Dans la pratique, le temps auquel on se situe correspond à la somme des poids des arêtes traversées jusqu'à présent, plus éventuellement des temps d'attente. Ainsi, si un sommet a la fenêtre $[56; 75]$, il faut que la somme des poids avant d'atteindre ce sommet soit comprise entre 56 et 75 et que l'on trouve un autre sommet où se rendre qui permettent de partir avant d'atteindre 75, c'est-à-dire que le temps d'arrivée minimal moins le temps du trajet soit inférieur à 75.

2 Métaheuristique choisie : Le recuit simulé

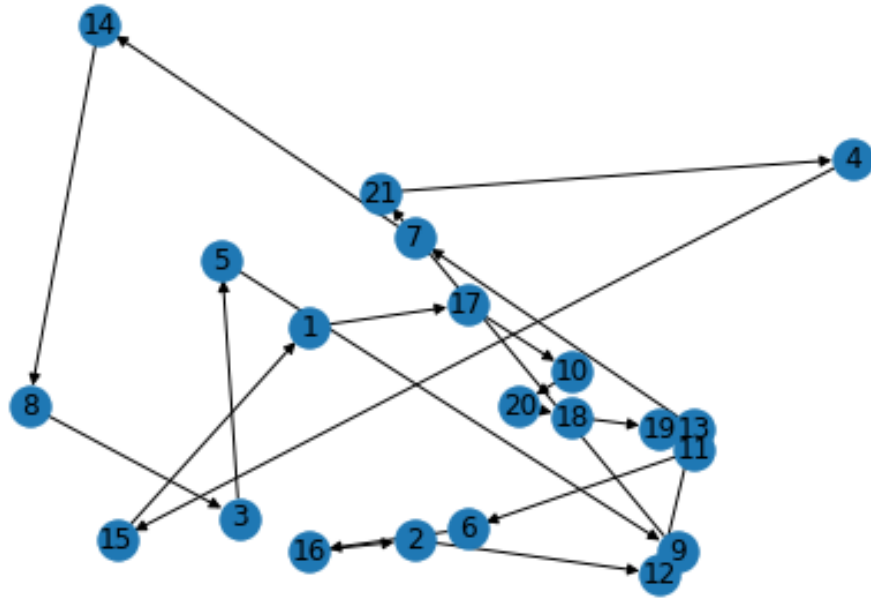
Nous avons choisi, afin de résoudre ce problème, la métaheuristique dite du recuit simulé. Le principe se base sur la physique et consiste en métallurgie à chauffer un métal puis à le faire refroidir lentement. Ici, on fixe une température de départ, puis on la fait décroître linéairement ou exponentiellement (on a choisi une décroissance exponentielle). Au tout départ, on prend une solution avec un agencement aléatoire des points. À chaque pas on actualise la température et on échange un nombre k de points entre eux (en pratique des places de sommets dans l'ordre de passage) ce qui actualise le score que l'on doit chercher à minimiser. On ne garde la nouvelle disposition que si elle permet de minimiser le score ou dans un cas aléatoire, si un tirage est en-dessous d'un certain seuil d'énergie (dépendant de la température donc). On répète l'opération un certain nombre de fois en repartant de la solution précédente.

Pour les choix du recuit simulé, nous avons décidé d'étudier 2 et 3-OPT qui consistent à échanger 2 ou 3 villes dans l'ordre déjà établi. Nous avons aussi regardé l'influence de la solution initiale sur le résultat.

3 Démarche de résolution

Nous devons réaliser des études sur trois solutions particulières. Pour des raisons de complexités, nous nous sommes d'abord concentrés sur l'instance avec le moins de villes. Elle comporte 21 villes.

FIGURE 1 – Représentation graphique de l'instance 1



Dans un premier temps, nous nous sommes concentré sur Nous avons cherché à améliorer itérativement plusieurs paramètres du recuit simulé : la fonction de score, puis la fonction de voisinage et enfin le choix des solutions aléatoires.

3.1 Choix du score

Dans un premier temps, nous avons étudié un cas simple où la fonction de voisinage est la permutation de 2 voisins. On choisit aussi une solution initiale au hasard parmi toutes les solutions (qu'elle soit faisable ou non). Dès les premiers benchmarks, nous avons remarqué que la construction de la solution faisable était déjà NP-Hard. Le score doit donc pouvoir symboliser à quelle point la solution est faisable ou non. Dans un premier temps nous avons défini un score naïf :

$$\text{score}_1 = nb_{\text{violation}} * 3 + \text{score}_{\text{solution}}$$

Ce premier score a le désavantage d'être par palier et il ne rend pas compte de l'avancée si on réduit le retard d'une ville. Nous avons alors étudié un second score :

$$\text{score}_{\text{retard}} = \sum \text{retard des individus}$$

Ce score est beaucoup plus lisse et permet de mieux rendre compte des avancées d'une solution lors du recuit simulé.

Nos test présents en partie 4 indiquent une amélioration notable en passant du premier score au second score. Avant de passer aux plus grandes instances, nous allons toujours étudier avec le premier score naïf mais nous choisissons le second score pour les instances plus grandes.

3.2 Choix de 2-OPT ou 3-OPT

Le second choix est la fonction de voisinages, nous avons voulu étudier le changement de 2-OPT à 3-OPT, autrement dit, est-ce que permuter 3 voisins au lieu de 2 est meilleur ? Le score obtenu pour la première instance étant déjà très faible, l'écart n'a pas été très visible. Pour des instances plus grosses, le résultat n'est pas clair : l'intervalle de confiance est plus élargi pour 3-OPT sans gain significatif en moyenne.

FIGURE 2 – nombre de violation pour 2OPT et un score de retard pour la première instance

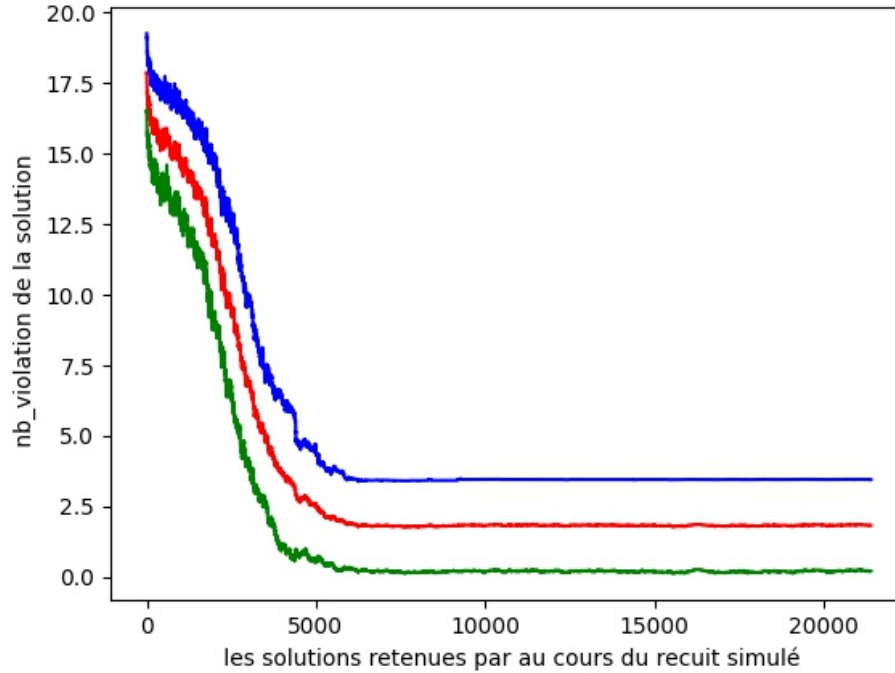
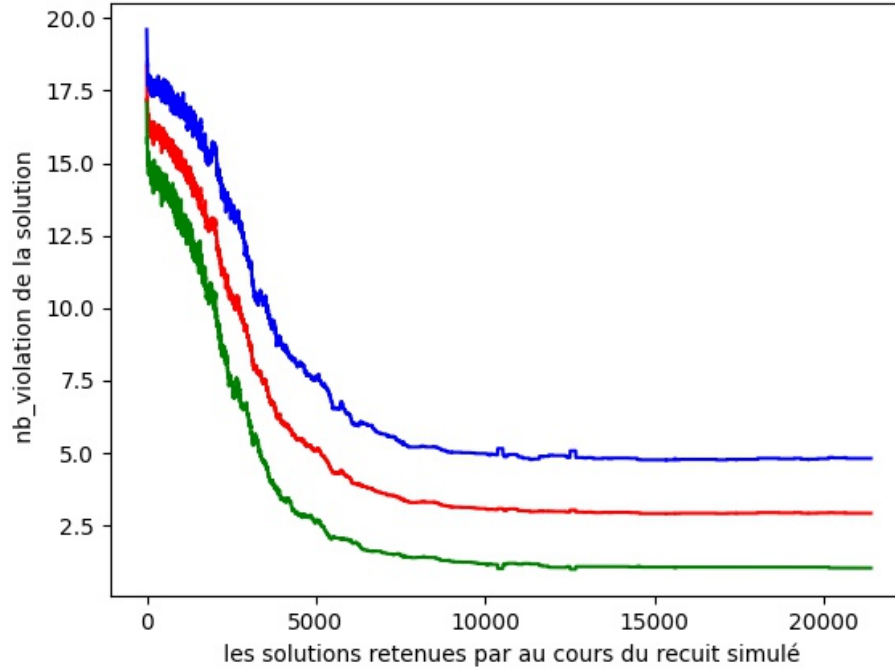


FIGURE 3 – nombre de violation pour 3OPT pour le même score



3.3 Choix des solutions initiales

Auparavant, la solution initiale était choisie aléatoirement, mais des calculs plus ou moins savants permettraient de choisir une solution initiale prometteuse. Nous n'avons étudié que 3 possibilités : des solutions aléatoires, des solutions où on trie selon la fenêtre de départ des instances et celle où on trie selon la fin des créneaux disponibles de visites. Pendant nos benchmarks, nous avons vu que le triage selon la fin ou le début de créneaux de visite ne change que peu la valeur des solutions obtenues.

Les solutions obtenues sont meilleures que si on avait choisi une solution aléatoire, mais la fonction de score semble signifier qu'on converge rapidement vers des minimums locaux dans le cas où la solution initiale est triée.

Pour résumer, nous avons remarqué que pour la première instance le choix d'une C.I triée selon le début de leur créneau avec un score de retard semble donner les résultats les plus prometteurs.

FIGURE 4 – nombre de violation pour des C.I initiales

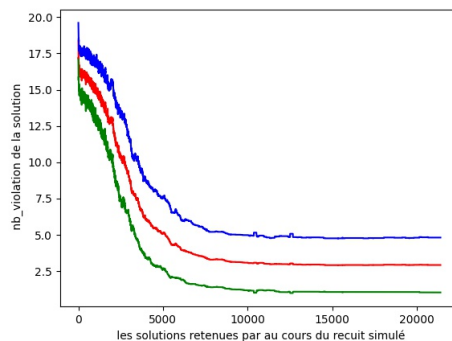
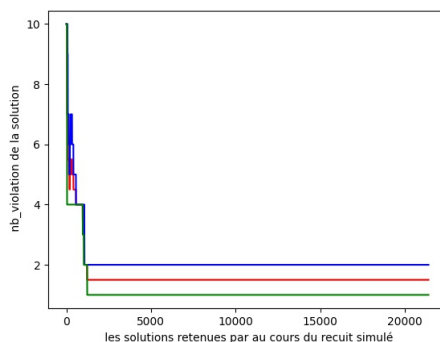


FIGURE 5 – Nombre de violation pour une C.I où les villes sont triée par le début de leur créneaux d'ouverture



4 Résultats sur les instances données

Nous regardons comment les résultats évoluent avec les deux autres instances. Nous n'avons pas beaucoup de résultat pour la troisième instance car le temps de calculs est beaucoup trop élevée

Nous avons été surpris par les courbes en fonction des Conditions initiales pour le modèle 2. En effet, les courbes obtenus sont très très semblable à la fin de l'algorithme quelque que soit la solution initiale choisie. Le problème vient peut-être du fait que la température était trop peu élevée ou que les solutions n'étaient pas assez bonnes. Les courbes sont bien différentes comme le souligne légèrement les conditions initiales...

La différence entre 2OPT et 3OPT semble aussi négligeable, les intervalles de confiances supérieures et inférieures sont quasiment égales...

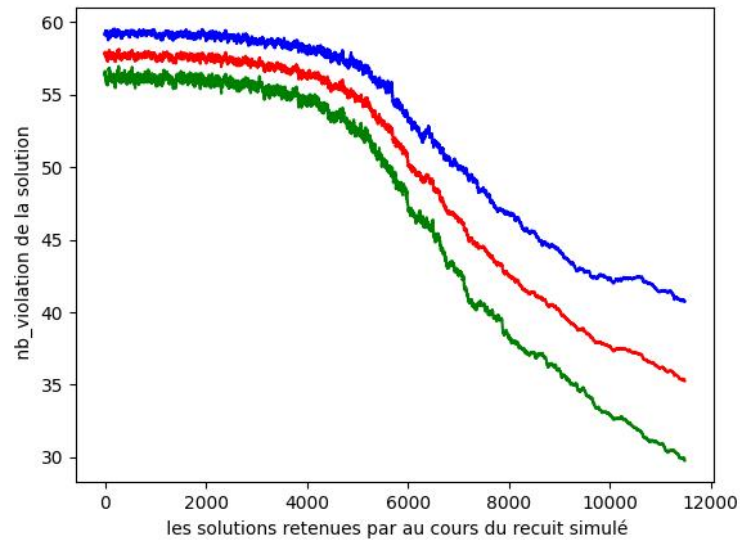
4.1 le cas du modèle 3

La encore, nous avons des résultats très semblables. Le changement de condition initiale ne semble pas avoir d'impact sur la qualité de la solution :

En fait, la condition initiale est très bonne de base mais il est dans un équilibre extrêmement instable ce qui nous ramène à une solution aléatoire quelconque. Une solution aurait été de baisser la température pour éviter de retomber dans des équilibres aussi mauvais. En faisant ça, on obtient les nouvelles courbes 14 et 15 :

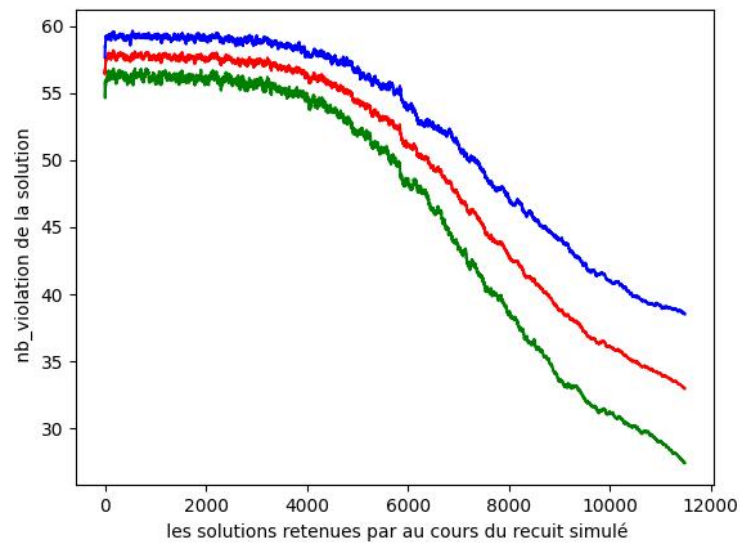
5 Conclusion

En conclusion, nous avons regardé la sensibilité du recuit simulé au nombre de villes échangées, à différentes valeurs de scores et à plusieurs conditions initiales. Visiblement, le score qui symbolise la somme des retards semblent d'être le meilleur score et le changement de 2OPT à 3OPT ne semble pas d'avoir d'incidence sur le problème. En revanche, la condition initiale semble primordiale au problème, sur tous nos graphiques, on remarque que si la température est trop élevée, la solution n'est pas idéale mais si la température est faible, la solution finale est du même ordre de grandeur que la condition initiale. Le choix de la condition initiale semble être le facteur prépondérant sur la réussite ou non du recuit simulé.



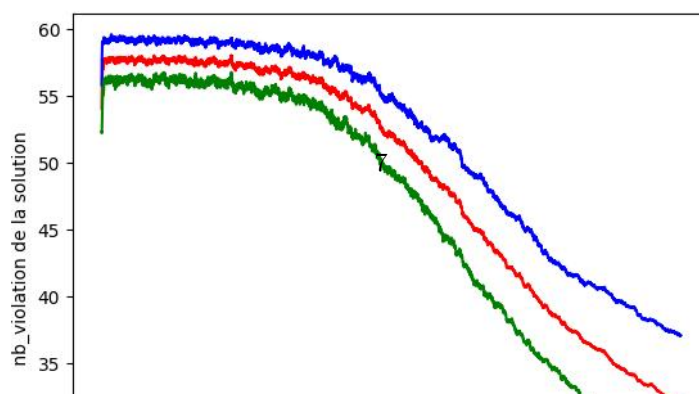
.5

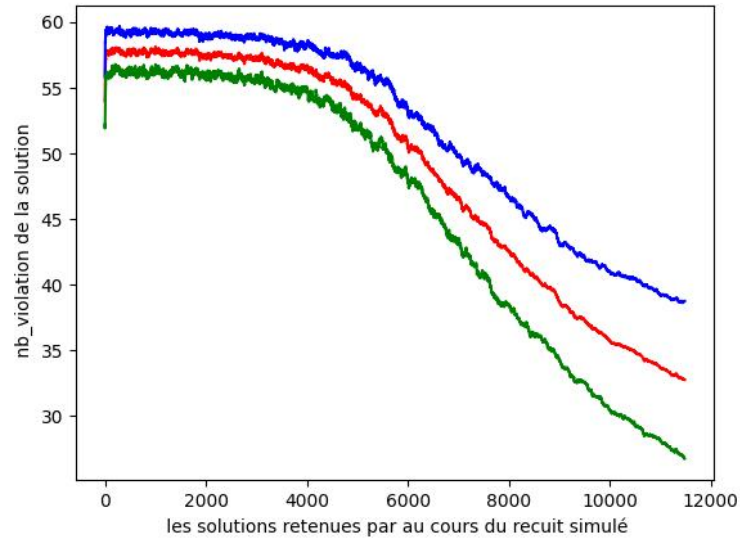
FIGURE 6 – cas où la solution aléatoire est aléatoire



.5

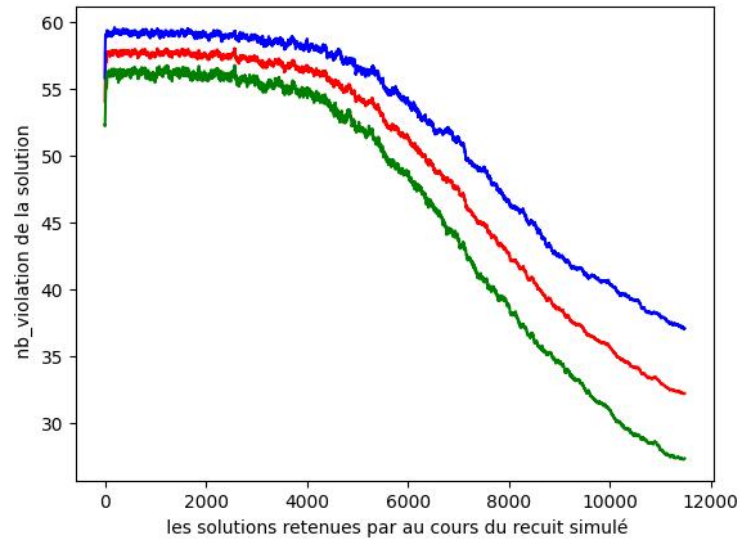
FIGURE 7 – cas où la C.I. est selon le début des créneaux





.5

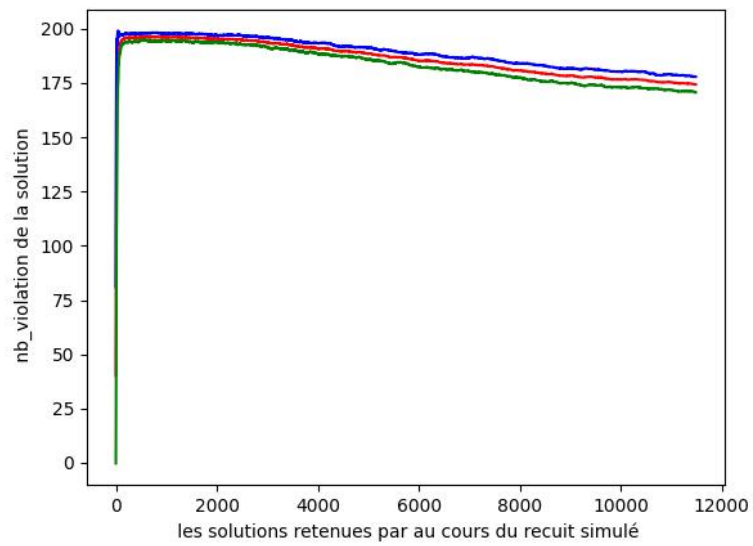
FIGURE 10 – cas 2OPT



.5

FIGURE 11 – cas 3OPT

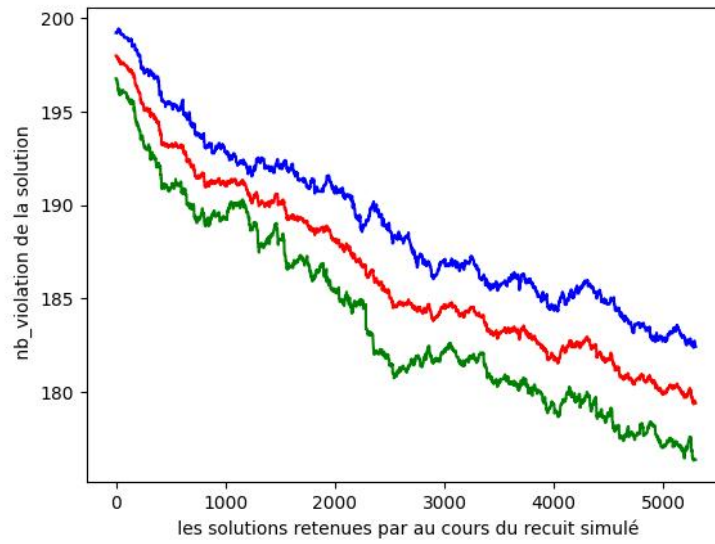
FIGURE 12 – Comparaison entre 2OPT et 3OPT pour l'instance 2



.5

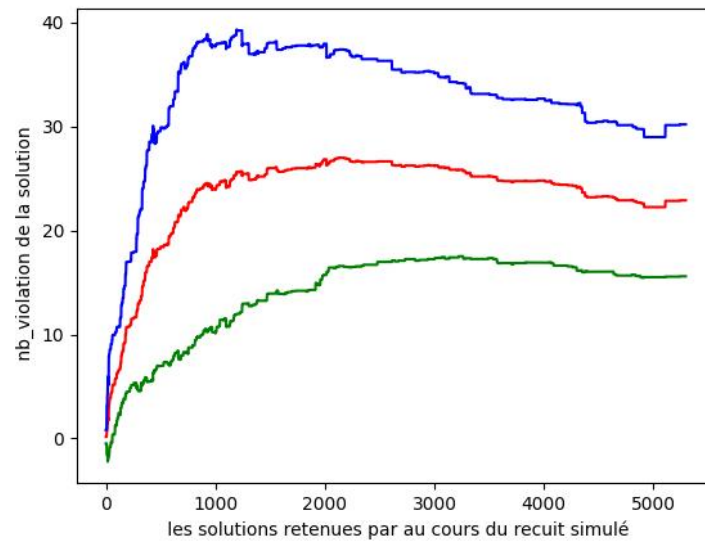
FIGURE 13 – cas où on trie selon la fin ou le début des fenêtres de visite

FIGURE 14 – Résultat pour l'instance3 pour les conditions initiales où on trie les individus



.5

FIGURE 15 – cas où la solution est aléatoire



.5

FIGURE 16 – cas où on trie selon le début de la fin des fenêtres de visite

FIGURE 17 – résultat pour différentes valeurs de l'instance 3